

# Supporting End-User Authoring of Alternate Reality Games with Cross-Location Compatibility

Sanjeet Hajarnis, Chinmay Barve, Devika Karnik, and Mark O. Riedl

School of Interactive Computing  
Georgia Institute of Technology  
{sanjeet, riedl}@gatech.edu; {cbarve, devika.karnik}@gmail.com

## Abstract

Alternate reality games (ARG) overlay a fictional story on top of the real world. Geo-location ties ARG storylines to specific locations in the real world. Unfortunately these games suffer from cross-location scalability limitation: games can only be played in very specific places. We describe location translation, a process whereby ARG locations are transformed to new locations through a combination of statistical information retrieval and greedy hill-climbing search.

## Introduction

Alternate Reality Games (ARGs) have emerged as a new genre of games. An ARG is an interactive narrative game that builds a superficial layer of reality above the actual reality. A typical ARG consists of a Puppet Master who designs the game and informs players of the unfolding of the story. With the advent of smart-phones with GPS, ARGs progressively make use of the actual world as the environment for which the game plays out.

Although ARGs are growing in popularity, there are certain limitations that they must overcome before being utilized by mainstream audiences. Firstly, supporting ARGs is effort-intensive on the part of Puppet Masters and confederates. This limitation can be overcome with technologies that push the role on the Puppet Master and confederate characters onto smart mobile devices. The WEQUEST game engine (Barve et al. 2010) is one such approach. The remaining effort is the authoring of game content. Secondly, mobile ARG game instances refer to real world geographical locations and landmarks for playing the game. Consequently, a particular instance of an ARG is fixed to a specific region of the real world, meaning it can only be played by a subset of people who live in proximity to that region without the intervention of a human puppet master. This poses a major constraint on the scalability of such geo-location based mobile ARG.

In this paper, we present an AI process whereby the locations referenced by an instance of a mobile ARG game are transformed to a new region so that any game instance can be played in any region. Our *location translation* process is couched in the WEQUEST system architecture that facilitates end-user authoring of new games and a game engine that can play novel game instances on geo-location enabled mobile devices. That is, any instance of a mobile ARG game instance can be automatically re-authored to work in a different place with minimal amount of supervision by human end-users.

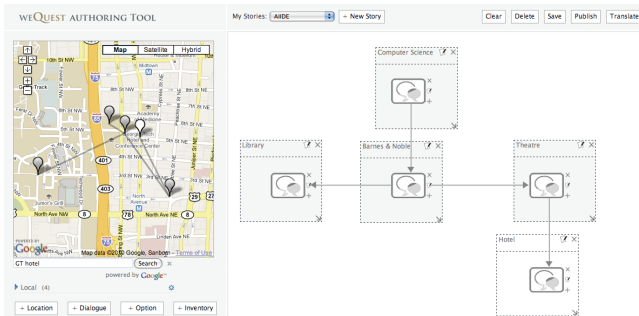
## The WEQUEST System

The WEQUEST (Barve et al. 2010) system consists of the following components, designed to overcome scalability issues that have historically prevented ARGs from mainstream adoption. A generic game engine runs on a geo-location enabled mobile device enables users to play any game modeled as a dependency graph of game content. An authoring tool enables users to author new games. The core contribution is the location translation process with which we can tackle the cross-location limitation of ARG storylines. The location translation process converts an instance of an ARG storyline set in one location to a new ARG storyline that can be played in a different location. Using this tool, authors could import a storyline with references to locations, for example in Atlanta, and translate it to play in another city.

## Story Representation

In WEQUEST, a game instance is a storyline, represented as a dependency graph where the nodes correspond to events, clues and reference specific GPS coordinates of the locations. Arcs between nodes represent dependencies that must be fulfilled for a particular event to fire.

A dependency graph is a basic technique for managing lock-and-key style game play. Unlike finite state machines, dependency graphs can support branching, partial ordering of events and parallelism. In our case, the branching is pre-authored into the dependency graph using the authoring



**Figure 1. A short story in the authoring tool.**

tool interface as opposed to a true Puppet Master altering the storyline.

## Location Translation

Location translation maps locations in the old storyline to analogous locations for a new area where the author intends to play. If a storyline cannot be played by a particular user because the storyline locations are not in user’s vicinity, the user can use the translation tool to obtain an analogous story that can be played in the user’s geographic area.

Location translation uses a combination of statistical information retrieval and search to find a reasonable mapping of old locations to new locations. In the current version of WEQUEST, each event node must be authored with tags indicating the salient features of that location (i.e. explaining their rationale behind choosing a particular location with respect to the storyline). The tags for each location in the original storyline are sent to the Google Maps™ API along with a GPS coordinate for the new area that the game is to be translated to.

A query to Google Maps™ API returns a set of candidate new locations for each old location. We assume that the candidate locations Google Maps™ retrieves are perfectly analogous to the original queries. The translation algorithm next attempts to find an optimal new storyline by selecting a candidate for each location in the original storyline. The optimal story path can be defined by its similarity to the original path in terms of (a) distance between adjacent locations, (b) total path distance, and (c) angles formed by any three consecutive locations to capture the original path shape. Note that, due to the possibility of branching storylines, locations can have more than two adjacent locations.

An exhaustive search would require a scan through all  $N^L$  possible path combinations, where  $N$  is the number of candidates per original location and  $L$  is the story length. We use greedy hill-climbing algorithm, guided by a heuristic (see next section) to find the set of retrieved candidate locations that should be swapped for the original locations. The greedy hill-climbing approach was selected for speed and because the subjective nature of “good” game; it is our experience that players don’t differentiate between optimal and sub-optimal translations as long as the locations are reasonably chosen, the distances are close

to the original, and zig-zags are avoided.

## Heuristic

The heuristic balances inter-location distance, total storyline distance, and shape. Inter-location distance captures the intended pacing of the author of the original storyline – short distances should remain short and long distances should remain long. Total storyline distance acts against the accumulation of inter-location error in the event that inter-location distances cannot be exactly matched.

We additionally include a shape metric in our heuristic. The heuristic checks for the angles between any three locations in Euclidean 2D space formed by locations’ latitudes and longitudes. Here we assume that earth is a 2D plane; this assumption is reasonable considering the proximity of locations. The shape heuristic was introduced to avoid zig-zag patterns in which the player has to move back and forth on the map to complete the storyline (unless the original storyline also contained zig-zags). In order to avoid zig-zag path formation, the heuristic favors larger angles between the three checkpoints. Unfortunately banning all acute angles would be detrimental for the algorithm because there could be stories that have a cycle that could require acute angles. In order to balance the heuristic, we normalize the distance and angle metric in our heuristic to 0.8 and 0.2 respectively.

## Limitations and Future Work

There are instances when the translation algorithm fails to find suitable translations for certain locations in the original storyline. Typically, the reason for this is the original author’s selection of tags with which to annotate locations. Tags that are too specific result in Google failing to find a match. Tags that are too general result in numerous irrelevant results. Future work involves overcoming the subjectiveness of author tag choice, using Machine Learning to automatically derive similarity between locations.

Location translation possesses the potential of solving the scalability problem for the location-based gaming. Our approach can generate a novel storyline consisting of new locations retrieved from Google Maps™ centered on a new region in which to play the game. With location translation, the limitation inherent in the fact that content in mobile ARG storylines is tied to the real world is reduced. The theoretical advantage is that end-users have access to a larger number of stories even though the stories were originally written to be played in remote places location.

## References

Chinmay Barve, Sanjeet Hajarnis, Devika Karnik, and Mark O. Riedl. 2010. WeQuest: A mobile alternate reality gaming platform and intelligent end-user authoring tool. *Proceedings of the 6th Conference on Artificial Intelligence for Interactive Digital Entertainment*.